EP 28609 ④

(12) UK Patent Application (19) GB (11) 2 335 125 (13) A

(43) Date of A Publication 08.09.1999

(21) Application No 9804683.2

(22) Date of Filing 04.03.1998

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s).
Ian Graham Holden

(74) Agent and/or Address for Service
M J Jennings
IBM United Kingdom Limited, Intellectual Property
Department, Mail Point 110, Hursley Park,
WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(51) INT CL^6
G06F 17/00

(52) UK CL (Edition Q)
H4T TBAX T109

(56) Documents Cited
EP 0513553 A2 WO 97/26608 A1 WO 94/28480 A1
US 5652714 A US 5515490 A

(58) Field of Search
UK CL (Edition P) H4T TBAX TBBA TBBN TBBX TCGX
TCHX
INT CL^6 G06F
ONLINE: WPI, INTERNET

(54) Abstract Title
SYSTEM AND METHOD FOR CREATING OR EDITING MULTIMEDIA PRESENTATION

(57) Provided are a method and a system for reducing re-rendering time for multimedia presentations following changes to input media components or to presentation-defining script information. The method involves automatically segmenting the presentation, determining which segments have changed as a result of changes to input media components or to script information such as frame location (time) or screen position, and then controlling subsequent rendering such that only changed segments are re-rendered. Animations are composited separately from still graphics and the segmentation and rendering is performed separately for each media type. Automated segmenting and determining of which parts of a presentation need re-rendering reduces rendering time and so speeds development while avoiding the effort and scope for human error of performing the same steps manually.

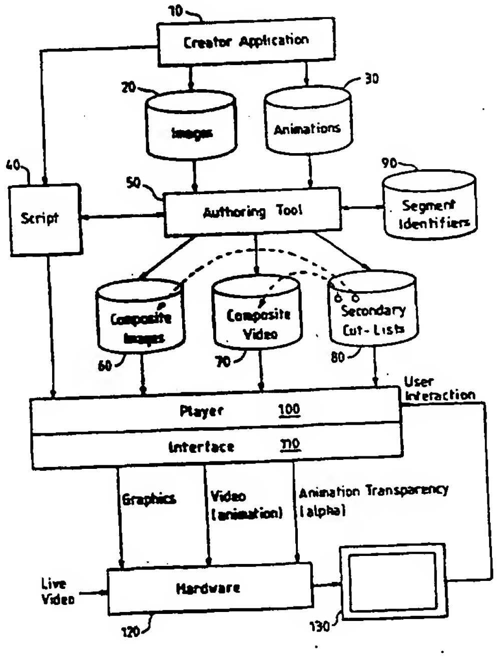

FIG. 1

GB 2 335 125 A

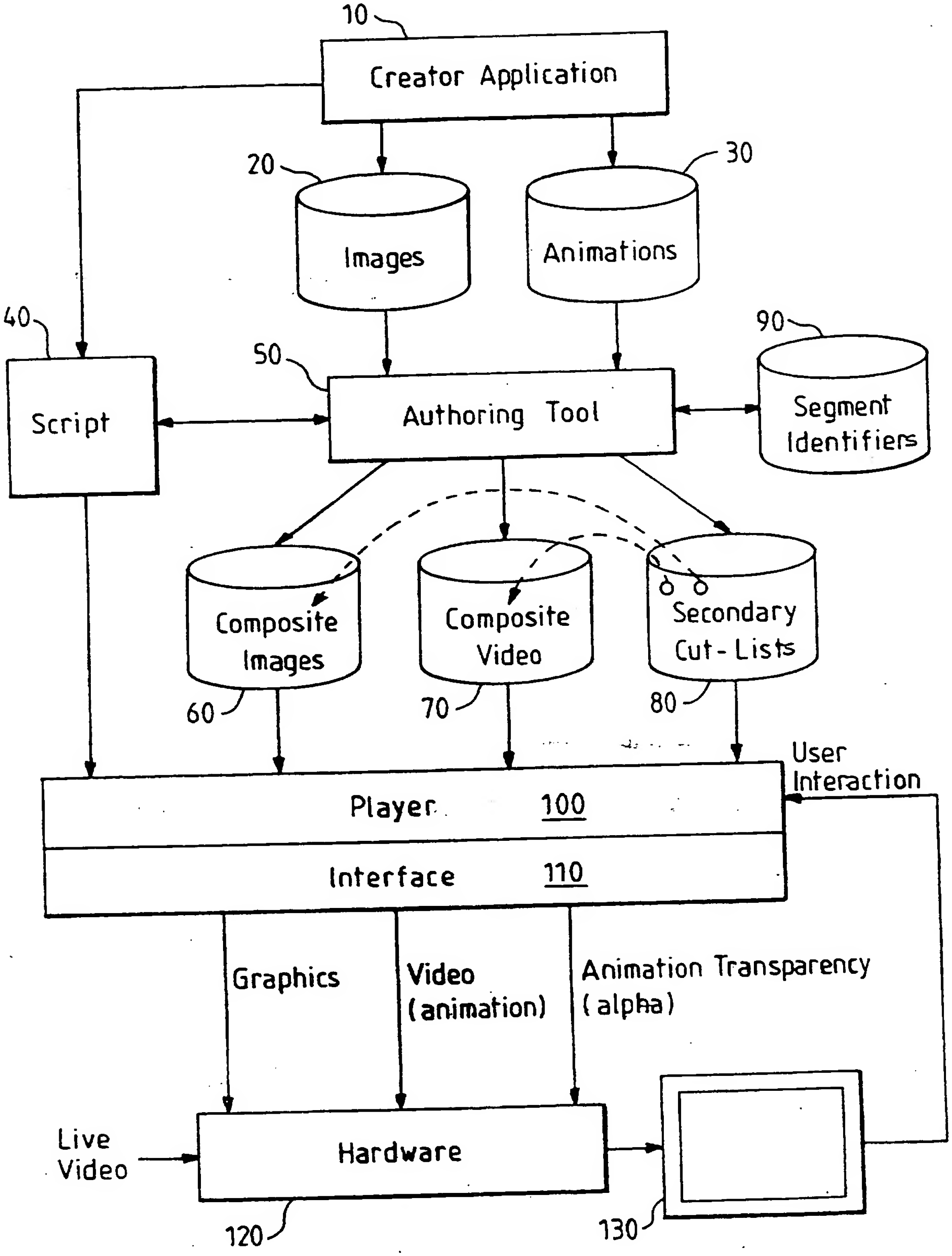
10
Creator Application
20
Images
30
Animations
40
Script
50
Authoring Tool
90
Segment Identifiers
Composite Images
Composite Video
Secondary Cut-Lists
60
70
80
User Interaction
Player 100
Interface 110
Graphics
Video (animation)
Animation Transparency (alpha)
Live Video
Hardware
120
130

FIG. 1

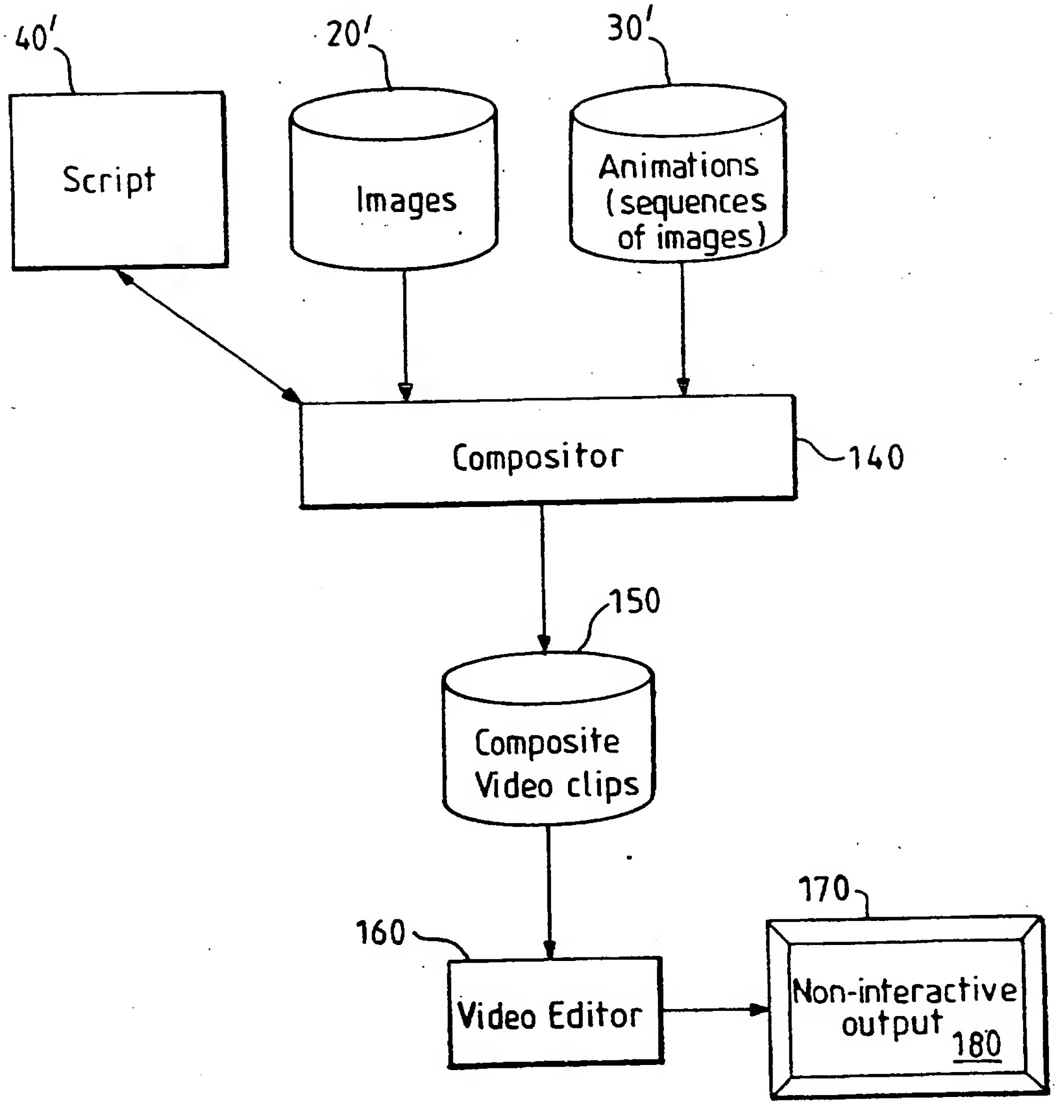

FIG. 2 PRIOR ART

```
                       time ---->


Source image A  AAAAAA....AA......AAA..
Source image B  ...BBBBBBBBBBBBBBBBBBBB
Source image C  .....C.........CCCCCCCC

Rendered images 11122344445566677788899
```

FIG. 3A

```
                         time ---->


Source animation P  ..PPPPPP....PP......PPP..
Source animation Q  ..QQQQQQQQQ..Q.QQQQQQQQQ.
Source animation R  .....RRRRRRR...RRRRRRR...

Rendered animation  :.111111111122.33333333
```

FIG. 3B

```
                      time ----->


Source image A  AAAAAA....AA......AAA..
Source image B  ...BBBBBBBBBBBBBBBBBBBB
Source image C  .....C.........CCCCCCCC

Rendered images 11122344445566677788899
With re-use     11122344442244455533355
```

FIG. 3C

```
*======================================================================
* 3 2 20 Wire Frame 2nd Church
*
PAGE :        03-02-020-WireFrame2ndChurch
*======================================================================
LIVEAREA  :        archi->03-00-010-ArchiSelect@Point
@start
*
* Archi is on screen till PointRight
* "Archaeology is the process of ..."
S       :       26mono48.wav
A       1:@PointRight-1      $APATH.archpoir\A_Lf_.tga PLACE(45,252) TRIM(1,1)
*
* 2nd church highlighted in red
G       1:@PointRight-1      $GPATH.graphics\grap3211.tga      PLACE(0,0)
* Church icon
G       SG:@end-1    $GPATH.graphics\grapchic.tga        PLACE(45,36)
WAIT :         SOUND
*
* Wire-Frame of 2nd Church in Red Rises
@PointRight
* "Here, the first layer exposed the ..."
S       :       27mono48.wav
A       N:D     $APATH.rise3220\2WR_.tga PLACE(0,0) TRIM(0,49)
G       N:@Fall-1       $APATH.rise3220\2WR_049.tga       PLACE(0,0)
G       SG:@Fall-1      $GPATH.graphics\grapchic.tga        PLACE(45,36)
*
A       SA:@Fall-1      $APATH.archpoir\A_Lf_.tga PLACE(45,252) TRIM(1,75) MIX
WAIT :         SOUND
*
*
*
* "In 1794 the church was sold and ..."
@Fall
S       :       28mono48.wav
* Wire-Frame of 2nd Church in Red Falls
A       N:D             $APATH.fall3220\2WF_.tga  PLACE(0,0) TRIM(0,49)
* 2nd church highlighted in red
G       N:@end-1       $GPATH.graphics\grap3120.tga       PLACE(0,0)
* Church icon
G       SG:@end-1      $GPATH.graphics\grapchic.tga        PLACE(45,36)
A       @Fall:@end-1 $APATH.archpoir\A_Lf_.tga PLACE(45,252) TRIM(1,75) MIX

WAIT :        $WaitBctwccnPage. SOUND->03-02-030-PlanView2ndChurch
*
@end
PAGE :        END
```

FIG. 4

```
File '03-02-020-WireFrame2ndChurch001.Ause' includes:
    A_Lf_.tga time=04Jun1997-09:47:20 pos=45,252 frames=11 trim=1,1 origin=0

File '03-02-020-WireFrame2ndChurch002.Ause':
    2WR_.tga time=29Oct1997-11:16:32 pos=0,0 frames=50 trim=0,49 origin=0
    A_Lf_.tga time=04Jun1997-09:47:20 mixA pos=45,252 frames=85 trim=1,75 origin=0

File '03-02-020-WireFrame2ndChurch003.Ause':
    2WF_.tga time=04Jul1997-20:06:08 pos=0,0 frames=50 trim=0,49 origin=0
    A_Lf_.tga time=04Jun1997-09:47:20 mixA pos=45,252 frames=85 trim=1,75 origin=0
```

FIG. 5A AVI files for animation segments

```
File '03-02-020-WireFrame2ndChurch001.Guse' includes:
    grap3211.tga time=09Oct1997-11:39:26 pos=0,0 frames=1
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1

File '03-02-020-WireFrame2ndChurch002.Guse' includes:
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1

File '03-02-020-WireFrame2ndChurch003.Guse' includes:
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1
    2WR_049.tga time=04Jul1997-20:06:08 pos=0,0 frames=1
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1

File '03-02-020-WireFrame2ndChurch004.Guse' includes:
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1

File '03-02-020-WireFrame2ndChurch005.Guse' includes:
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1
    grap3120.tga time=09Oct1997-11:47:16 pos=0,0 frames=1
    grapchic.tga time=19Oct1997-16:06:30 pos=45,36 frames=1
```

FIG. 5B Graphics segments

SYSTEM AND METHOD FOR CREATING OR EDITING MEDIA PRESENTATIONS

Field of Invention

The present invention relates to authoring and editing of media presentations such as composite video and animation presentations.

Background

Multimedia presentation authoring and editing can now be performed on a personal computer, for example using commercially available software products such as Adobe Illustrator to aid the creation of images and animations and using media compositor software such as Adobe Premiere (both from Adobe Systems Inc). Separate software is available for playback of the composite video. Hardware which is available for supporting playback of a plurality of composite video streams includes Matrox Electronic Systems Limited's DigiSuite video and graphics editing and mixing PCI-boards, interoperating with playback software implementing, for example, Microsoft Corporation's ActiveMovie media-streaming architecture and application programming interface (API). These PCI-boards may be connected within a conventional personal computer (PC).

Such a PC media editing environment can be used to build up linear video streams from live or stored video clips, graphics (still images) and animation files. Rendered video streams may be played back and mixed together in real time by the mixer cards, including handling layering of the graphics and video layers, to produce a single output video presentation. However, the rendering process (i.e. the process of retrieving image and animation files from storage and arranging these components in time and position into video streams for transmittal or display) is very slow. It is typical for rendering of each minute of video to take up to an hour (approximately one frame every 2 seconds).

This rendering time is a significant problem for authors of complex interactive presentations, which are made up from many animations and graphic images, since the author will need to make a great many changes to the composition layout, timings, etc, as the presentation is developed and will want to see the results of those changes as soon as possible. The changes may require many re-renderings of long video streams which can be very time consuming, such that the author experiences a long delay before they can view their updated presentation.

Large and complex systems are available which can handle re-rendering of images in real-time, but such systems are extremely expensive and are not adaptable to the PC environment.

Prior art PC-based authoring tools have attempted to maximise the performance of their rendering algorithms, but have not reduced rendering times sufficiently to meet the requirements of authors of interactive animated presentations.

Summary of Invention

In a first aspect, the present invention provides a multimedia presentation authoring tool for use in the creation of a multimedia presentation from input media components and script information relating to the media components, the authoring tool including:

a renderer for rendering the media components in accordance with the script information to create an output multimedia presentation;

a segmenter for assigning segment identifier information to presentation segments of the output multimedia presentation in accordance with predefined segmentation criteria and for storing the assigned segment identifier information in association with the multimedia presentation; and

a controller, responsive to changes to the input media components and/or the script information, for determining which of the presentation segments are affected by the changes and controlling the renderer so as to render only the affected segments.

The input media components which are operated on by the authoring tool typically include static image files, animation files and video recordings. The input media components, when operated on by the authoring tool, preferably create at least one separate segmented output data stream for each media type, these separate output streams then being mixed together by a mixer card at playback time (and possibly also mixed with live video) to produce a single video presentation. The "script" information is presentation-defining data such as component position information, start and end frames for each component, layering information (i.e. whether animations overlay graphics and both overlay video, for example), and mixing information (i.e. how the different streams interact - such as transparency characteristics). The term "script" is used here to reflect that the information is a description of when and where components appear and their characteristics and interactions - it may or may not be written in a script language.

A multimedia authoring tool according to the invention can significantly reduce the time taken to re-render a presentation after changes are made, without relying on the author to make decisions about which parts of an output video stream should be re-rendered. Automating the determination of which parts of a presentation need to be re-rendered, by segmenting the output presentation and identifying which segments are affected by changes to input components and/or their associated script information, both avoids user effort and achieves an authoring tool which is less prone to human error.

In the prior art, a "cut-list" is known to be a description of a simple sequence of video clips (media components) including start and end frames (script information) for the clips which make up the sequence. A multimedia presentation tool according to a preferred embodiment of the present invention similarly uses a "primary cut-list" or input script information which defines sequences of input components (video clips, graphics and animations) to generate a plurality of output media streams, but the input script for the graphics and animation streams typically include more varied data than for standard video editing applications. The input script ("primary data may include start and end frames, the number of frames, media component position information, layering and mixing information, as mentioned above. The presentation tool of the present invention then automatically generates, from the information within the primary cut-list or script, "secondary cut-lists" defining a segmentation of each of the output composite graphics and animation streams of the multimedia presentation. These secondary cut-lists are advantageous for reducing subsequent re-rendering times.

The secondary cut-lists are preferably a set of parallel cut-lists including a first cut-list for still images, which first cut-list comprises a description of a sequence of full screen composite images, and one or more cut-lists for animations, which comprise(s) a description of a sequence of full screen composite animations. A secondary cut-list is typically not required for camera-originated video sequences which do not require re-rendering.

The authoring system maintains the secondary cut-lists in association with the output presentation, such that the segment divisions within each cut-list remain identifiable and useable to identify which segments of the output presentation are affected by changes to input components. In a preferred embodiment, segment identifier files including all of the information required to re-construct a segment from its component parts, and timestamps of the time of last modification of each component within the segment, are generated and stored together with each of the segment media files. When any changes are made to input media components or script information and rendering is subsequently required, a new identifier file is generated for each composite still image segment or each composite animation segment of the presentation and the new identifiers are compared with stored identifiers to determine which segments have changed. If there are differences, the changed segments are re-rendered and the new identifiers are stored.

The predefined segmentation criteria controlling the segmenting are preferably related to both the script information and the nature of the media components. According to a preferred embodiment, a graphic images cut-list includes a defined segment for each different static full screen composite image; whereas an animation cut-list includes a defined segment for each sequence which begins with the start of a first animation and either ends with the completion of that animation or, if another animation overlaps the first animation sequence, ends upon completion of the last of the series of concurrent (i.e. partially overlapping) animations. Additionally, segment size criteria may be applied to the segmentation. These criteria will be described in more detail later with reference to Figure 3.

The invention according to a second aspect provides a method for generating a multimedia presentation from input media components and script information relating to said media components, the method including the steps of:

rendering the media components in accordance with the script information to generate an output multimedia presentation;

assigning segment identifier information to presentation segments of the output multimedia presentation in accordance with predefined segmentation criteria, and storing the assigned segment identifier information in association with the multimedia presentation; and in response to changes to the input media components and/or the script information, determining which of the presentation segments are affected by the changes and controlling rendering of the media components such that only the affected segments are re-rendered.

The invention may be implemented in a stand-alone data processing system including the authoring tool, a presentation playback application, and video stream mixing hardware. Alternatively, the invention may be implemented in a system within a client-server environment in which an interactive application requires downloading of files to a client system from a server system.

Brief Description of Drawings

Preferred embodiments of the present invention will now be described in more detail, by way of example, with reference to the accompanying drawings in which:

Figure 1 is a schematic representation of the major components of a PC-based multimedia authoring system suitable for implementing the invention;

Figure 2 is a schematic representation of a prior art PC-based video editing system;

Figures 3a, 3b and 3c show examples of segmentation of video streams in accordance with preferred embodiments of the invention;

Figure 4 shows an example extract of an input script suitable for analysis by the authoring tool according to an embodiment of the present invention;

Figure 5A shows example identifier files for segments of a composite animation sequence for a presentation defined by the script of Figure 4; and

Figure 5B shows example identifier files for segments of a sequence of composite graphics for the presentation defined by the script of Figure 4.

Detailed Description of Preferred Embodiments

An example PC-based video and graphics editing and mixing system according to the invention is shown schematically in Figure 1. A presentation editing and rendering tool 50 (referred to hereafter as the "authoring tool") is shown receiving input images 20 and animations 30 (sequences of images) from a multimedia creation application program 10. A further input to the authoring tool 50 is a presentation-defining script 40 created using the media creation application 10 and editable by both the creation application 10 and the authoring tool 50. The authoring tool 50 and the creation application 10 could be integrated as a single multimedia authoring application program, but for the purposes of the description of the preferred embodiment they are described as separate components for clarity.

A user of the system generates images and animations using the creation application 10 and writes a presentation definition or script which defines which image and animation should appear where and when (i.e. at which frames of the presentation it starts and ends). The creation application 10 preferably includes a graphical user interface (GUI) to aid script creation, but the script creation may require the user to have computer programming skills as in the prior art. The user typically also defines in the script the layering order of the different media types (for example, the animation plane or layer overlays the graphics layer which overlays video) and mixing characteristics such as the extent of transparency (i.e. the extent to which parts of the video show through the animations layer, etc).

The authoring tool 50 creates composite full screen image files 60 from the input images and script information, and creates composite full screen video clips 70 from the input animations and script information relating to the animations, outputting a separate data stream for each of the animations and the graphics images. The user-defined script includes an instruction as to which layer of the presentation each input component should be composited into (i.e. generally all still images in one output stream for one layer of the presenation, all animations in another stream for a second layer, and possibly transparency information for the animations in a further separate stream if the encoding to achieve the output streams requires this). Of course, the identification of the input media types and their separation into respective media-type-specific composite streams may be, in alternative embodiments of the invention, an automated analysis of the input components rather than being explicitly flagged in the script.

The computations to achieve this rendering of input components to create the composite images 60 and composite video 70 involve, as is known in the art, analysis of the position of each component (in time and X,Y position on screen, including identifying component boundaries), analysis of the relative layering of animations and still images, and analysis of mixing factors such as the degree of transparency, all of which information is preferably defined in the script information.

In addition, the authoring tool according to the present invention generates "secondary" cut-lists 80 (i.e. output cut lists additional to the input script) logically segmenting the presentation and describing the positioning, start and end frames, and other presentation-defining characteristics of each of the components forming each segment of the composite image sequences and composite video clips. This generation of "secondary" cut-lists 80 describing segments of the composited video streams will be described in detail below.

The composite image files 60 and animation sequences 70, together with their associated input script information 40 and secondary cut-list information 80, are then input to a player component 100. The player 100 manages real-time transmission of the various composite image and animation streams, received from the authoring tool 50, to video editing and mixing hardware 120. An interactive control terminal 130 enables the user to select a particular video sequence and thus to control the player component 100.

The video editing and mixing hardware 120 may, for example, comprise one or more of Matrox Electronic Systems Limited's DigiSuite PCI-cards integrated within a conventional personal computer connected to an input/output terminal 130.

The system described with reference to Figure 1 is distinguished from prior art video editing systems in a number of ways. A prior art system is shown schematically in Figure 2. Note that a compositor application program 140, such as Adobe Systems Inc's Premiere software, typically does not keep input images and animation sequences separate but composites them together into a single video output 150 in accordance with script information 40' which is defined by the software user. The composite video clips are then passed to a non-linear editor 160 to control playback and display as a single non-interactive television output 180 on a display terminal 170.

In the prior art system of Figure 2, any changes which require the output video 150 to be re-rendered are dealt with by re-rendering the whole presentation. Alternatively, a user could manually create an output cut-list which splits the output presentation into time segments and then manually determine which segments have changed and which are unchanged. Such manual creation of a cut-list by analysing a video presentation would be time consuming, and so any savings in re-rendering time would be wasted, and would also be likely to introduce errors or at least would be unlikely to achieve optimal segmentation of the presentation.

A system implementing the present invention, such as is shown in Figure 1, improves on prior art rendering of video, graphics and animation files by implementing the following additional steps:

1. Automatically dividing the rendered presentation into small self-contained sequences or "segments".

2. Giving each segment an identifier 90 that identifies all the information needed to re-construct the segment from its component parts including a time stamp showing when each component was last modified. Each segment identifier becomes an item in a cut-list 80 that defines the whole video stream for that media type.

3. Optimising re-rendering by using the segment identifiers and their timestamps to determine which segments have changed. Only the changed segments are then re-rendered.

After a change is made to any component, or to the timing or position of any component within the presentation, the re-rendering is optimised by restricting re-rendering to only those segments which have changed. This significantly reduces the time taken to re-render, and automating this avoids the effort and the potential for human error that arise if the author were to be relied upon to make decisions about which parts of an animation stream should be re-rendered.

Dividing the presentation into segments

The division of the presentation into small segments can be performed entirely by the segmenter of the authoring tool 50, so the author need not be aware of how the presentation is divided up. In the preferred embodiment, the small segments into which the presentation is divided are suitable for play back in real time by the Matrox DigiSuite card set using the Microsoft ActiveMovie interface.

A cut-list or script describing where and when input components should appear in a presentation is a useful tool for video editing where a number of video clips are to be played in sequence. However, an author of interactive multimedia applications typically works with a larger number of source media elements (images, animations, video clips etc.) than other video editors and the media components are presented over time in a complex way, often with many components on the screen at the same time. The multimedia author, ideally, needs the system to take care of how the various streams and layers are rendered and split into cut-lists.

Video cards such as the Matrox DigiSuite card set can support a number of graphic and animation (or video) cut-lists input to them via the Microsoft ActiveMovie interfaces. Given a presentation script, which defines how the various source images and animations are to be presented, the authoring system of the present invention renders these into a number of graphic and animation streams defined by secondary cut-lists.

An example extract of an input script defined using the creation application 10 is shown in Figure 4. This extract represents a page of a presentation (i.e. a section of time of the presentation, which has a logical beginning and end) including sound clips, animations and still graphics. At the start of the page, a soundclip file '26mono48.wav' and the animation sequence 'A_Lf_.tga PLACE(45,252) TRIM(1,1)' are to be played, and graphics file 'grap3211.tga PLACE(0,0)' is to be displayed.

The animation descriptor statement at the start of the example script of Figure 4 is explained below:

```
A      1:@PointRight-1   $APATH.archpoir\A_Lf_.tga PLACE(45,252) TRIM(1,1)
↑            ↑                             ↑
Control  Start and end frames        File location and definition
```

The control indicates the type of statement, which may be an action statement such as an instruction to present a particular type of media component (such as 'A' for present animation, 'G' for graphic, 'S' for sound clip) or to wait for a specific event or time ('WAIT'), or may be a comment ('*'), a configuration setting, a variable definition or a label ('@...'). The second element in the statement gives the start and end of this animation sequence within the frames of this page of the presentation. The start and end frames can both be identified by frame numbers or using location-simplifying labels (such as '@start' or '@PointRight-1' meaning the frame before location '@PointRight' which is defined in the example script) or special codes ('SA' to mean start of previous animation, for example). The third element is a variable giving the drive and directory location and definition for this animation component. The definition includes the animation component file name ('A_Lf_.tga'), its X,Y position on a display ('PLACE(45,252)' means screen position 45 pixels from the left and 252 pixels from the top), and a specific set of images within that named animation file (in this case 'TRIM(1,1)' means show image 'A_Lf_001.tga' only).

Similarly, the statement:

```
A     N:D    $APATH.rise3220\2WR_.tga PLACE(0,0) TRIM(0,49)
```

means show animation frames '2WR_000.tga' through '2WR_049.tga' positioned at top left of screen.

The statement "show graphic 'grap3211.tga' at top left of screen (position X=0,Y=0) between frame 1 of the page and the frame before frame location '@PointRight'" appears in the script as:

```
G     1:@PointRight-1   $GPATH.graphics\grap3211.tga PLACE(0,0)
```

Typically, for multimedia presentations using animations, still graphics, and live source video from a camera or source video from a laser disk, for example, the visual elements will be rendered from their individual components into three or more parallel streams that are played and mixed in real time:

1. Source video;

2. Graphics stream containing still images including transparency information;

3. Encoded animation video stream (e.g. MJPEG) excluding transparency information;

4. Encoded animation video stream (e.g. MJPEG) containing only transparency information.

These streams are arranged in logical presentation layers ("priorities"), normally from back to front respectively. The transparency information in each layer defines how much of the layer behind will show through (ranging from fully opaque to fully transparent and mixed according to one of a number of different mixing methods - such as method 'MIX' which involves simple averaging of pixel values taking account of percentage transparency values). So, at the back is source video from a camera (live) or VCR (recorded), for example; overlaid on the live video are still images; and overlaid on these are animations. Note that the animation layer requires two streams to provide transparency information, as encoding such as MJPEG and MPEG2 do not include transparency. Another stream is therefore used to define the transparency element of the animations.

To optimise re-rendering, each of the composite media streams (excluding live video which needs no re-rendering) is defined by a cut-list 80 which is a definition of a sequence of small parts called cut-list segments. As each cut-list segment is a self contained media file 90, it can be re-rendered independently of the other cut-list segments. The segmented output stream 60 formed from the static images comprises a sequence of targa (TGA) format files, one TGA file for each segment; whereas the segmented output stream 70 formed from input animation components comprises a sequence of MJPEG-encoded AVI files - two files for each segment as explained above. The conversion of input images and animations into TGA files and AVI files uses the facilities of the editing hardware 130 (for example, Matrox Electronic Systems Limited's DigiSuite card set).

Simply dividing a stream at arbitrary points, such as a division into clips of equal duration, is not advisable. Any segmentation policy which is sufficiently simple to carry out manually is unlikely to satisfy the conflicting requirements of minimizing playback performance loss and substantially reducing re-rendering time, especially in view of the added constraint that still graphics, animations and input video are typically all composited together in prior art systems. Optimal dividing of an interactive presentation into segments requires consideration of:

1. the ability to replay the segmented presentation smoothly;

2. the requirement for start and stop points where interaction can occur; and

3. the likelihood of re-rendering particular sequences.

According to the preferred embodiment of the present invention, the authoring tool 50 parses through the script 40 identifying all action statements and uses their types (A, G, S, etc) to determine which statements relate to media components that are to be composited together. The authoring tool then analyzes the start and end frame information within statements which relate to a given media type to identify overlaps in time, and applies a segmentation policy which divides the presentation data for each media type into segments.

Automated creation of a graphic images cut-list 80 uses the authoring tool 50 to analyze the input script 40 and to compose from the script 40 a cut-list segment identifier 90 for each static full screen in the graphic layer. Each appearance and removal of a component image which changes the composite full screen image defines a transition between cut-list segments. This is shown in Figure 3a.

In the example of Figure 3a, the source images A, B and C are displayed through time in such a way that 9 full screen composite images (cut-list segments) need to be rendered for use in the graphic cut-list.

An animation cut-list is created by the authoring tool 50 by starting each new cut-list segment where one or more animations start, and ending it when all concurrent animations have either finished, or are on their last frame. This is shown in Figure 3b.

In this example, the source animations P, Q and R are played through time in such a way that three composite full screen animation sequences form segments in the animation stream cut-list which need to be rendered separately.

This method of creating an animations cut-list is suitable for typical multimedia applications using many short animations. It ensures that all component animations play in their entirety, each from a single cut-list segment. This reduces the risk of play-back delays or errors, due to system performance limitations which are often stretched to the limit at the transition points between cut-list segments.

Where long animations are used (for example, over one minute long), re-rendering times are advantageously further reduced by making secondary splits. These secondary splits are made either at the start or end of a source animation depending on which achieves a more balanced segmentation. In the preferred embodiment, if an animation segment would exceed 100 seconds (2500 frames), then the mid point of the segment is found and then a scan is performed to identify the animation start or end point which is closest to this mid point and that start or end point is then taken as the point to split the segment. Splits are not made if they would create any very short cut-list segments of, for example, 10 frames or less. There is an inherent trade-off between the desirability to maximize the number of segments for optimised rendering at authoring time and to minimize the number of segments for improved performance at playback, and so limiting to segments of 25 frames (1 second of playback time) or more and 1500 frames (60 seconds) or less is preferred where possible.

If each cut-list segment is given a unique identifier 90 (as will be described in more detail below), identical cut-list segments can be easily identified to avoid duplicating the re-rendering of identical cut-list segments. For example, if we assume that the source media A, B and C in the graphic cut-list example of Figure 3a referred to above are always presented in the same positions on the screen, cut-list item 2 is identical to 5, 3 = 8, 4 = 6 and 7 = 9, so these segments may be re-used reducing the number of segments requiring re-rendering from 9 to 5. This is shown in Figure 3c.

Now we have divided the presentation into sequences of full sreen images and animations, we simply need a method of knowing which of these has been changed since last rendered. Segment Identifiers 90 provide this capability.

Building the segment identifiers

Each segment identifier includes all the information needed to reconstruct the segment from it's component parts - all of the information needed to fully define the script information for each input component of a segment. A time stamp for each component part is also included to indicate when it was last modified.

For each composite still image, the following information on each component part is retrieved from the input script and then included in the composite image's identifier:

1. file name

2. last modified date & time

3. position

4. mix method

For each composite animation sequence, the following information about each component animation is retrieved from the script and included in the composite animation segment's identifier:

1. file name

2. last modified date & time

3. position

4. mix method
5. start frame of the segment for this animation

6. first frame of animation

7. last frame of animation

8. number of frames

A particular example of a series of identifier files 90 for the segments of the presentation defined by the script of Figure 4 is shown in Figures 5A and 5B. Note that the identifier files include collections of animation definition files (Figure 5A) or collections of graphic definition files (Figure 5B) retrieved from the script. The particular selection of which definition files are collected together to form a presentation segment is determined by the segmentation policy as described above. The identifier files for all segments of a particular media type output stream comprise the secondary cut-list 80 for that output stream.

Rendering

Whenever a composite still image or animation sequence is rendered, its identifier is saved with it as a separate file. The precise location of each image file or animation file and the associated identifier file does not have to be carefully managed in the preferred embodiment of the invention, since the associations between files are identifiable by means of a common filename prefix of the form:

ClipName.avi	for an animation file
ClipName.use	for an identifier file
ClipName.a-avi	for an alpha file representing transparency characteristics

Whenever rendering is required, each composite still image or animation sequence (i.e. each segment of either the graphics data stream or the animation data stream) is considered in turn. First its new identifier 90′ is built using information obtained from the presentation-defining script file 40. The new identifier 90′ is then compared with the saved identifier 90 from the previous rendering. If there is any difference, the sequence is re-rendered, and a new identifier 90′ saved. The first time a presentation is rendered the same process is performed but, since there are no stored identifier files to compare with, all identifier parameters for all segments are considered to have changed and so all segments are rendered.

As discussed previously, conventional mixing cards 120 handle real-time mixing of rendered data streams (each representing full screen image, animation, or video files) to produce a composite video presentation ready for playback.

The timestamps are used to ensure that re-rendering of a segment occurs in cases where the only change which has been made to the segment is to the content of a media component without changing the component's file name, position or any other other cut-list-defined characteristics. For example, an author may change the colour of an image of a car from blue to green without any change to the cut-list information other than the last modification timestamp, and this will necessitate re-rendering of the segment which includes that changed image.

An additional improvement can be made to prior art multimedia authoring, which improvement is particularly advantageous for creation of interactive multimedia presentations using the automated segmenting of the present invention. A problem arises with interactive applications in that it can be difficult to ensure that a presentation stops at a precise intended frame of the presentation to allow user interaction, or in response to user interaction, or merely to allow playback of an audio recording to complete. Of course, stopping one frame too early or one frame too late can mean that the composite image which is displayed to the user is not the correct one. This could mean that the user is unable to understand what interactive steps are required, or that he or she is unaware that any interactive action is required at all, or that the displayed image is irrelevant to the voiceover.

The solution to this problem implemented in a preferred embodiment of the present invention is to replicate any frame which is required to be a wait point and then to reset the stop frame to be a mid-frame within the series of replica frames (for example, add 10 replica frames to any frame at a point in the application at which maintenance of the currently displayed image is required and then redefine the wait point to be the fifth frame in the sequence of replica frames). This multiplication of frames at any wait point is, in the preferred embodiment, an automated response to a "WAIT" instruction within the presentation defining script. It is implemented by the authoring tool 50. Wait points are natural animation segment boundaries and so this frame replication step for stop points is a modification of the segmentation step described earlier.

In an alternative embodiment of the present invention, an authoring tool implements segmenting and rendering of only changed segments as described above but the authoring tool has no editing capability. Any required changes to a presentation are made by the creation application which then provides images, animations and the presentation-defining script to the authoring tool ready for rendering. The function of the authoring tool according to this embodiment is thus limited to converting input script information and associated images and animations into parallel video streams (each representing either a sequence of composite images or a sequence of composite animations) and generating the associated secondary cut-lists which define the segment boundaries and information for recreating each segment.

The segment definition function of the present invention may also be used in an alternative embodiment of the invention in a networked environment in which a user interacts with an interactive application via a client computer system which responds by requesting files from a server computer connected across the network. This may be implemented by companies wishing to provide interactive Web pages for access by Internet users. The segmentation and enablement of identification of changed segments according to the present invention is used by the client system to identify what segments of a presentation are required to be displayed to the user in response to the user's interaction, and then only the required segments are downloaded to the client system rather than downloading the complete interactive presentation. This minimizes the size of rendered multimedia files that must be retrieved from the server - reducing network traffic and the workload of the server computers and avoiding over-large interactive presentations filling the memory of the client system.

CLAIMS

1. A multimedia presentation authoring tool for use in the creation of a multimedia presentation from input media components and script information relating to the media components, the authoring tool including:

a renderer for rendering the media components in accordance with the script information to create an output multimedia presentation;

a segmenter for assigning segment identifier information to presentation segments of the output multimedia presentation in accordance with predefined segmentation criteria and for storing the assigned segment identifier information in association with the multimedia presentation; and

a controller, responsive to changes to the input media components and/or the script information, for determining which of the presentation segments are affected by the changes and controlling the renderer so as to re-render only the affected segments.

2. An authoring tool according to claim 1, adapted to generate new segment identifier information in response to changes to the input media components and/or script information, and to compare said new segment identifier information with said stored segment identifier information to determine which segments are affected by the changes.

3. An authoring tool according to claim 1 or claim 2, wherein the segment identifier information is generated from said script information relating to the media components and includes the position of each media component within the segment and a timestamp of the time of last modification of each component within the segment.

4. An authoring tool according to claim 3, adapted to create a multimedia presentation from input animations, wherein the renderer is adapted to generate one or more composite video streams from said input animations and wherein the segment identifier information for a segment of said one or more composite video streams includes an identification of the start frame within the segment and the number of frames, and the first and last frame of the animation, for each animation within the segment.

5. An authoring tool according to claim 1, adapted to create a multimedia presentation from input animations and input graphics, wherein the renderer is adapted to generate one or more composite video streams from said input animations and to generate at least one separate composite data stream from said input graphics, and wherein the segmenter is adapted to assign segment identifier information separately for the output data streams for the animations and graphics.

6. An authoring tool according to claim 5, wherein the segmenter is adapted to assign separate segment identifier information to any portion of the multimedia presentation which comprises a series of concurrent animations which series begins with the start of a first animation and ends with the last frame of the series of concurrent animations.

7. An authoring tool according to claim 6, wherein the segmenter is adapted to split segments which comprise a series of concurrent animations if the segment would exceed a preset maximum number of frames.

8. An authoring tool according to any one of claims 5 to 7, wherein the segmenter is adapted to assign separate segment identifier information to each different composite full-screen graphic image within the graphics data stream.

9. A data processing system for use in the creation of multimedia presentations from input media components and script information relating to the media components, the system including:

an authoring tool according to claim 5;

a mixer for mixing said composite video streams to produce an output video stream.

10. A method for generating a multimedia presentation from input media components and script information relating to said media components, the method including the steps of:

rendering the media components in accordance with the script information to generate an output multimedia presentation;

assigning segment identifier information to presentation segments of the output multimedia presentation in accordance with predefined segmentation criteria, and storing the assigned segment identifier information in association with the multimedia presentation; and

in response to changes to the input media components and/or the script information, determining which of the presentation segments are affected by the changes and controlling rendering of the media components such that only the affected segments are re-rendered.

11. A method according to claim 10, including the steps of:

generating new segment identifier information in response to changes to the input media components and/or script information;

comparing said new segment identifier information with said stored segment identifier information to determine which segments are affacted by the changes.

12. A method according to claim 10 or claim 11, including automatically replacing any frame of said multimedia presentation which is defined as a stop frame with a series of replica frames of said stop frame, and redefining the stop frame as a mid-frame of the series of replica frames.

13. A method for reducing rendering time for a multimedia presentation formed from input media components and script information relating to said media components, the method including the steps of:

rendering the media components in accordance with the script information to generate an output multimedia presentation;

automatically assigning segment identifier information to presentation segments of the output multimedia presentation in accordance with predefined segmentation criteria, and storing the assigned segment identifier information in association with the multimedia presentation; and in response to changes to the input media components and/or the script information, determining which of the presentation segments are affected by the changes and controlling rendering of the media components such that only the affected segments are re-rendered.

The Patent Office

21

Application No:	GB 9804683.2	**Examiner:**	R F King
Claims searched:	All	**Date of search:**	24 April 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4T[TBAX, TBBA, TBBN, TBBX, TCGX, TCHA, TCHX]

Int Cl (Ed.6): G06F

Other: ONLINE: WPI; INTERNET

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	EP 0 513 553 A2	[COMMODORE] See abstract	1, 10 and 13
"	WO 94/28480 A1	[IMAGINE MULTIMEDIA] See abstract	"
"	WO 97/26608 A1	[VICOM MULTIMEDIA] See abstract	"
"	US 5,652,714 A	[APPLE] See abstract	"
"	US 5,515,490 A	[XEROX] See abstract	"

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		